

1 11. The method of claim 8, wherein the resolved entry of the task class
2 mirror table associated with the class is used in cases where testing for class
3 initialization is unneeded but access to a task-private part of the class is required
4 when the class has been loaded but not fully initialized.

1 12. The method of claim 6,
2 wherein task class mirror tables associated with classes that have an empty
3 initialization function have a single entry per task; and
4 wherein the single entry per task is the initialized entry for that task.

1 13. The method of claim 12, further comprising:
2 upon loading the class that has the non-empty initialization function by the
3 task, creating the task class mirror object that holds the task private representation
4 of the class;
5 setting the task class mirror object's state to loaded; and
6 assigning the task class mirror object's pointer to a resolved entry of the
7 task class mirror table associated with the class for that task.

1 14. The method of claim 13,
2 wherein the task class mirror table is arranged so that the resolved entry
3 and the initialized entry for the task are separated by half of a total number of
4 entries in the task class mirror table; and
5 wherein the byte-offset to the resolved entry can be computed from the
6 byte-offset to the initialized entry for a same task by adding a size, expressed in
7 number of bytes, of half the total number of entries in the task class mirror table.

15. The method of claim 14, wherein the resolved entry of task class mirror tables associated with classes that have the non-empty initialization function is used when accessing a task-private part of the class without testing for class initialization is necessary and the task has loaded but not fully initialized the class.

16. The method of claim 12, further comprising:
upon loading of the class that has the empty initialization function by the task, creating the task class mirror object that holds the task private representation of the class;
setting the task class mirror object's state to fully initialized; and
assigning the task class mirror object's pointer to the initialized entry of the task class mirror table associated with the class for that task.

17. A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method to efficiently realize class initialization barriers in a multitasking virtual machine, wherein class loading always takes place before class initialization, and wherein a class initialization barrier guarantees that a class is initialized before the class is first used by a program, comprising:
associating a shared runtime representation of the class with a task class mirror table that comprises at least one entry per-task, including an initialized entry, for a plurality of tasks, wherein each entry holds either a null pointer value or a non-null pointer to a task class mirror object, wherein all entries of a task mirror table that hold a non-null pointer value and that are associated with a same task hold a pointer to a same task class mirror object, wherein the task class mirror object holds a task private representation of the class for that task;

14 using the initialized entry of a task in the task class mirror table to
 15 determine whether this task has initialized the class associated with the task class
 16 mirror table; and
 17 accessing the task class mirror object associated to a particular task.

1 18. The computer-readable storage medium of claim 17,
 2 wherein each task is associated with a unique integer value;
 3 wherein the unique integer value is used to compute a byte-offset from a
 4 beginning of task class mirror tables that can be used to retrieve from the
 5 initialized entry of any task class mirror table the pointer to the task class mirror
 6 object; and
 7 wherein a computed byte-offset to the initialized entry is stored in a
 8 descriptor of a plurality of threads executing on behalf of a corresponding task.

1 19. The computer-readable storage medium of claim 18, the method
 2 further comprising:
 3 creating the task class mirror table and associating the task class mirror
 4 table with the shared runtime representation of the class upon creation of the
 5 shared runtime representation of the class; and
 6 setting all entries of the task class mirror table to the null pointer value.

1 20. The computer-readable storage medium of claim 19, the method
 2 further comprising:
 3 examining the initialized entry of the task in the task class mirror table
 4 associated with the class in order to determine if that task has initialized the class,
 5 wherein the byte-offset to the initialized entry from the beginning of the task class
 6 mirror table is obtained from the descriptor of a thread performing an examination